Adaptive Cruise Control:
Hybrid, Distributed, and Now Formally Verified

Sarah M. Loos **André Platzer** **Ligia Nistor**

JUNE 2011
CMU-CS-11-107

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA

This material is based upon work supported by National Science Foundation under NSF CAREER Award CNS-1054246 and Grant Nos. CNS-0926181, CNS-0931985, CNS-1035800, CNS-1035813, and ONR N00014-10-1-0188. The first author was supported by an NSF Graduate Research Fellowship. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution or government.
For proofs and interactive car system simulations, see `http://www.ls.cs.cmu.edu/dccs/` online.
A conference version of this report has appeared at FM [LPN11a].

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUN 2011		**00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Adaptive Cruise Control: Hybrid, Distributed, and Now Formally Verified	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Carnegie Mellon University,School of Computer Science,Pittsburgh,PA,15213**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Car safety measures can be most effective when the cars on a street coordinate their control actions using distributed cooperative control. While each car optimizes its navigation planning locally to ensure the driver reaches his destination, all cars coordinate their actions in a distributed way in order to minimize the risk of safety hazards and collisions. These systems control the physical aspects of car movement using cyber technologies like local and remote sensor data and distributed V2V and V2I communication. They are thus cyber-physical systems. In this paper, we consider a distributed car control system that is inspired by the ambitions of the California PATH project, the CICAS system, SAFESPOT and PReVENT initiatives. We develop a formal model of a distributed car control system in which every car is controlled by adaptive cruise control. One of the major technical difficulties is that faithful models of distributed car control have both distributed systems and hybrid systems dynamics. They form distributed hybrid systems, which makes them very challenging for verification. In a formal proof system, we verify that the control model satisfies its main safety objective and guarantees collision freedom for arbitrarily many cars driving on a street even if new cars enter the lane from on-ramps or multi-lane streets. The system we present is in many ways one of the most complicated cyber-physical systems that has ever been fully verified formally.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**28**	

Keywords: Distributed car control, multi-agent systems, highway traffic safety, formal verification, distributed hybrid systems

Abstract

Car safety measures can be most effective when the cars on a street coordinate their control actions using distributed cooperative control. While each car optimizes its navigation planning locally to ensure the driver reaches his destination, all cars coordinate their actions in a distributed way in order to minimize the risk of safety hazards and collisions. These systems control the physical aspects of car movement using cyber technologies like local and remote sensor data and distributed V2V and V2I communication. They are thus cyber-physical systems. In this paper, we consider a distributed car control system that is inspired by the ambitions of the California PATH project, the CICAS system, SAFESPOT and PReVENT initiatives. We develop a formal model of a distributed car control system in which every car is controlled by adaptive cruise control. One of the major technical difficulties is that faithful models of distributed car control have both distributed systems and hybrid systems dynamics. They form distributed hybrid systems, which makes them very challenging for verification. In a formal proof system, we verify that the control model satisfies its main safety objective and guarantees collision freedom for arbitrarily many cars driving on a street, even if new cars enter the lane from on-ramps or multi-lane streets. The system we present is in many ways one of the most complicated cyber-physical systems that has ever been fully verified formally.

1 Introduction

Because of its societal relevance, numerous parts of car control have been studied before [CCB+07, DHO06, DCH08, DCH07, HCG03, HC02, HESV91, Ioa97, JKI99, HTS04, Shl04, SFHK04, Var93, WMML09, CT94, JR03, AAWB10, BSBP03]. Major initiatives have been devoted to developing next generation individual ground transportation solutions, including the California PATH project, the SAFESPOT and PReVENT initiatives, the CICAS-V system, and many others. Chang et al. [CCB+07], for instance, propose CICAS-V in response to a report that crashes at intersections in the US cost $97 Billion in the year 2000. The promise is tempting. Current uncontrolled car traffic is inefficient and has too many safety risks, which are caused, e.g., by traffic jams behind curves, reduced vision at night, inappropriate reactions to difficult driving conditions, or sleepy drivers. Next generation car control aims to solve these problems by using advanced sensing, wireless V2V (vehicle to vehicle) and V2I (vehicle to roadside infrastructure) communication, and (semi)automatic driver assistance technology that prevents accidents and increases economical and ecological efficiency.

Yet, there are several challenges that still need to be solved to make next generation car control a reality. The most interesting challenge for us is that it only makes sense to introduce any of these systems after its correct functioning and reliability has been ensured. Otherwise, the system might do more harm than good. This is the formal verification problem for distributed car control, which we consider in this paper.

What makes this problem particularly exciting is its practical relevance. What makes it particularly challenging is its complicated dynamics. Distributed car control follows a hybrid dynamics, because cars move continuously along differential equations and their behavior is affected by discrete control decisions like when and how strongly to brake or to accelerate and to steer. It is in the very nature of distributed car control, however, to go beyond that with *distributed* traffic agents that interact by local sensing, broadcast communication, remote sensor data, or cooperative networked control decisions. This makes distributed car control systems prime examples of what are called *distributed hybrid systems*. In fact, because they form distributed cyber-physical multi-agent systems, the resulting systems are distributed hybrid systems regardless of whether they are built using explicitly distributed V2V and V2I network communication infrastructure or just rely on the distributed effects of sensor readings about objects traveling at remote locations (e.g., laser-range sensors measuring the distance to the car in front).

Cars reach maneuvering decisions locally in a distributed way. Is the global dynamics that emerges from the various local choices safe? What can a car assume about other cars in its maneuver planning? How do we ensure that multiple maneuvers that make sense locally do not cause conflicts or collisions globally? Formal verification of distributed hybrid systems had been an essentially unsolved challenge until recently [Pla10].

Our main contribution is that we develop a distributed car control system and a formal proof that this system is collision-free for arbitrarily many cars, even when new cars enter or leave a multi-lane highway with arbitrarily many lanes. Another contribution is that we develop a proof structure that is strictly modular. We reduce the proof to modular stages that can be verified without the details in lower levels of abstraction. We believe the principles behind our modular structure and verification techniques are useful for other systems beyond the automotive domain. Further

contributions are:

- This is the first case study in distributed hybrid systems to be verified with a generic and systematic verification approach that is not specific to the particular problem.
- We identify a simple invariant that all cars have to obey and show that it is sufficient for safety, even for emergent behavior of multiple distributed car maneuvers.
- We identify generic and static constraints on the input output parameters that any controller must obey to ensure that cars always stay safe.
- We demonstrate the feasibility of distributed hybrid systems verification.

2 Related Work

Car control is a deep area that has been studied by a number of different communities. The societal relevance of vehicle cooperation for CICAS intersection collision avoidance [Shl04] and for automated highway systems [HCG03, Ioa97] has been emphasized. Horowitz et al. [HTS04] proposed a lane change maneuver within platoons. Varaiya [Var93] outlines the key features of an IVHS (Intelligent Vehicle/Highway System). A significant amount of work has been done in the pioneering California PATH Project. Our work is strongly inspired by these systems, but it goes further and sets the groundwork for the modeling and formal verification of their reliability and safety even in distributed car control.

Dao et al. [DCH07, DCH08] developed an algorithm and model for lane assignment. Their simulations suggest [DCH08] that traffic safety can be enhanced if vehicles are organized into platoons, as opposed to having random space between them. Our approach considers an even more general setting: we not only verify safety for platoon systems, but also when cars are driving on a lane without following platooning controllers. Hall et al. [HC02] also used simulations to find out what is the best strategy of maximizing traffic throughput. Chee et al. [CT94] showed that lane change maneuvers can be achieved in automated highway systems using the signals available from on-board sensors. Jula et al. [JKI99] used simulations to study the conditions under which accidents can be avoided during lane changes and merges. They have only tested safety partially. In contrast to [DCH07, DCH08, HC02, CT94, JKI99], we do not use simulation but formal verification to validate our hypotheses.

Hsu et al. [HESV91] propose a control system for IVHS that organizes traffic in platoons of closely spaced vehicles. They specify this system by interacting finite state machines. Those cannot represent the actual continuous movement of the cars. We use differential equations to model the continuous dynamics of the vehicles and thus consider more realistic models of the interactions between vehicles, their control, and their movement.

Stursberg et al. [SFHK04] applied counterexample-guided verification to a cruise control system with two cars on one lane. Their technique can not scale to an arbitrary number of cars. Althoff et al. [AAWB10] use reachability analysis to prove the safety of evasive maneuvers with constant velocity. They verify a very specific situation: a wrong way driver threatens two autonomously driving vehicles on a road with three lanes.

Wongpiromsarn et al. [WMML09] verify safety of the planner-controller subsystem of a single autonomous ground vehicle. Their verification techniques restrict acceleration changes to fixed and perfect polling frequency, while our model of an arbitrary number of cars allows changes in acceleration at any point in time, with irregular sensor updates.

Damm et al. [DHO06] give a verification rule that is specialized to collision freedom of traffic agents. To show that two cars do not collide, they need to manually prove eighteen verification conditions. Lygeros and Lynch [LL98] prove safety only for one deceleration strategy for a string of vehicles: the leading vehicle applies maximum deceleration until it stops, while at the same time, the cars following it in the string decelerate to a stop. The instantaneous, globally synchronized reaction of the cars is an unrealistic assumption that we do not make in our case study. Dolginova and Lynch [DL97] verify that no collisions with big relative velocity can occur when two adjacent platoons do a merge maneuver. This does not prove the absence of small relative velocity collisions, nor the behavior of 3 platoons or when not merging. In contrast to the manual semantic reasoning of [DHO06, LL98, DL97], our techniques follow a formal proof calculus [Pla10], which can be mechanized. In the case studies analyzed by [LL98, DL97] safety is proved only for a particular scenario, while our modular formal proofs deal with the general case. In our case study, the cars have more flexibility and an arbitrary number of control choices.

Unlike [DHO06, SFHK04, AAWB10, WMML09], we prove safety for an arbitrary number of cars. Our techniques and results are more general than the case-specific approaches [DHO06, SFHK04, AAWB10, LL98, DL97, WMML09], as we prove collision-freedom for any number of cars driving on any finite number of lanes. None of the previously cited papers have proved safety for distributed car control in which cars can dynamically enter the highway system, change lanes, and exit.

3 Preliminaries: Quantified Differential Dynamic Logic

Distributed car control systems are distributed hybrid systems, which we model by *quantified hybrid programs* (QHPs) [Pla10]. QHPs are defined by the grammar (α, β are QHPs, θ a term, i a variable, f a function symbol, and H a formula of first-order logic):

$$\alpha, \beta \quad ::= \quad \forall i : C\ f(i) := \theta \mid \forall i : C\ f(i)' = \theta \,\&\, H \mid f(i) := * \mid ?H \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^*$$

The effect of *quantified assignment* $\forall i : C\ f(i) := \theta$ is an instantaneous discrete jump assigning θ to $f(i)$ simultaneously for all objects i of type C. Usually i occurs in θ. The effect of *quantified differential equation* $\forall i : C\ f(i)' = \theta \,\&\, H$ is a continuous evolution where, for all objects i of type C, all differential equations $f(i)' = \theta$ hold *and* (written $\&$ for clarity) formula H holds throughout the evolution (the state remains in the region described by H). Usually, i occurs in θ. Here $f(i)'$ is intended to denote the derivative of the interpretation of the term $f(i)$ over time during continuous evolution, not the derivative of $f(i)$ by its argument i. For $f(i)'$ to be defined, we assume f is an $\mathbb{R}$-valued function symbol. The effect of the random assignment $f(i) := *$ is to non-deterministically pick an arbitrary number or object (of type the type of $f(i)$) as the value of $f(i)$.

The effect of *test* $?H$ is a *skip* (i.e., no change) if formula H is true in the current state and *abort* (blocking the system run by a failed assertion), otherwise. *Non-deterministic choice* $\alpha \cup \beta$ is for alternatives in the behavior of the distributed hybrid system. In the *sequential composition* $\alpha; \beta$, QHP β starts after α finishes (β never starts if α continues indefinitely). *Non-deterministic repetition* α^* repeats α an arbitrary number of times ≥ 0.

For stating and proving properties of QHPs, we use *quantified differential dynamic logic* QdL [Pla10] with the grammar:

$$\phi, \psi \ ::= \ \theta_1 = \theta_2 \mid \theta_1 \geq \theta_2 \mid \neg\phi \mid \phi \wedge \psi \mid \forall i : C \ \phi \mid \exists i : C \ \phi \mid [\alpha]\phi \mid \langle\alpha\rangle\phi$$

In addition to all formulas of first-order real arithmetic, QdL allows formulas of the form $[\alpha]\phi$ with a QHP α and a formula ϕ. Formula $[\alpha]\phi$ is true in a state ν iff ϕ is true in all states that are reachable from ν by following the transitions of α; see [Pla10] for details.

4 The Distributed Car Control Problem

Our approach to proving safety of a distributed car control system is to break the verification into modular pieces. In this way, we simplify what would otherwise be a very large and complex proof. The ultimate result of this paper is a formally verified model of any straight stretch of highway on which each car is following adaptive cruise control. On any highway, there will be an arbitrary number of lanes and an arbitrary number of cars, and the system will change while it runs when cars enter and leave the highway.

This would be an incredibly complex system to verify if we were to tackle it at this level. Each lane has a group of cars driving on it. This group is constantly changing as cars weave in and out of surrounding traffic. Each car has a position, velocity, and acceleration, and must obey the laws of physics. On top of that, in order to ensure complete safety of the system, every car must be certain at all times that its control choices will not cause a collision anywhere else in the system at any time in the future.

These issues are compounded by the limits of the sensory and communications networks. On a highway that stretches hundreds of miles, we could not hope for any car to collect and analyze real-time data from every other car on the interstate. Instead, we must assume each car is making decisions based on its local environment, e.g., within the limitations of sensors, V2V and V2I communication, and real-time computation.

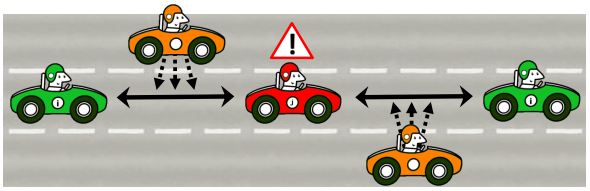

Figure 1: Emergent highway collision risk

Additionally, once you split your system into reasonably local models, it is still difficult to reason about how these local groups of cars interact. For example, consider a local group of three cars for a lane change maneuver: the car changing lanes, and the two cars that will be ahead and behind it. It is tempting to signal the car ahead to speed up and the car behind to slow down in order to make space for the car changing lanes.

This is perfectly reasonable on the local level; however, Fig. 1 demonstrates a problem that appears when we attempt to compose these seemingly safe local cases into a global system. Two cars are attempting safe and legal lane changes simultaneously, but the car which separates the merging cars is at risk. The car in the middle simultaneously receives requests to slow down and speed up. It cannot comply, which could jeopardize the safety of the entire system.

To avoid complex rippling cases that could result in a situation similar to the one in Fig. 1, we organize our system model as a collection of hierarchical modular pieces. The smallest piece consists of only two cars on a single lane. We present a verification of this model in Sect. 5 and build more complex proofs upon it throughout the paper.

In Sect. 6, we prove that a lane with an arbitrary number of cars driven by any distributed homogeneous adaptive cruise control system is safe, assuming the system has been proved safe for two cars. We generate our own verified adaptive cruise control model for this system, but, due to the modular proof structure, it can be substituted with *any* implementation-specific control system which has been proved safe for two cars.

The verification of this one lane system, as well as the verification we present in Sect. 8 for a highway with multiple lanes, will hold independently with respect to the adaptive cruise control specifications. In Sect. 7, we look at the local level of a multi-lane highway system. We verify the adaptive cruise control for a single lane, where cars are allowed to merge in and out of the lane. Finally in Sect. 8, we compose the lane systems verified in Sect. 7 to provide a full verification of the highway system.

5 Local Lane Control

The local car dynamics problem that we are solving is: we have two cars on a straight lane that can accelerate, coast or brake and we want to prove that they will not collide. This system contains complex physical controls as well as discrete and continuous dynamics, thus, is a hybrid system. Once the model for the local problem is verified, we will use it in a compositional fashion to prove safety for more complicated scenarios, such as multiple cars driving on a lane or on parallel lanes. We can apply modular composition because we have structured the models in a hierarchical order, we have found the right decomposition of the sub-problems and we have identified the right invariants.

5.1 Modeling

We develop a formal model of the local car dynamics as a QHP. Each car has state variables that determine how it operates: position, velocity, and acceleration. For follower car f, x_f represents its position, v_f its velocity, and a_f its acceleration (similarly for leader car ℓ).

The continuous dynamics for f are described by the following differential equation system: $x_f' = v_f,\ v_f' = a_f$. This is the ideal-world dynamics that is adequate for a kinematic model of longitudinal lane maneuvers. The rate with which the position of the car changes is given by x_f', i.e., the velocity. The velocity itself changes continuously according to the current acceleration a_f. We do not assume permanent control over the acceleration, but tolerate delays since sensor

readings are not available continuously, control decisions may need time, and actuators may take time to react. For simplicity, though, we still assume that, once set, the acceleration a_f takes instant effect. We assume a global limit for the maximum acceleration and we denote it by $A \geq 0$. We assume that all cars have an emergency brake with a braking power between a maximum value B and a minimum value b, where $B \geq b > 0$. The two values have to be positive, otherwise the cars cannot brake. They may be different, however, because we cannot expect all cars to realize exactly the same emergency braking power and it would be unrealistic to build a system based on the assumption that all reactions are equal.

In Fig. 2, we see that leader ℓ brakes unexpectedly at time $t1$ with its maximum braking power, $-B$. Unfortunately, f did not follow ℓ at a safe distance, and so when sensor and network data finally inform f at time $t2$ that ℓ is braking, it is already too late for f to prevent a collision. Although f applies its full braking power, $-b$, at time $t2$, the cars will inevitably crash at time $t3$. The same problem can happen if ℓ brakes with $-b$ and f brakes with $-B$. This example shows that control choices which look good early on can cause problems later. Adding cars to the system amplifies these errors.

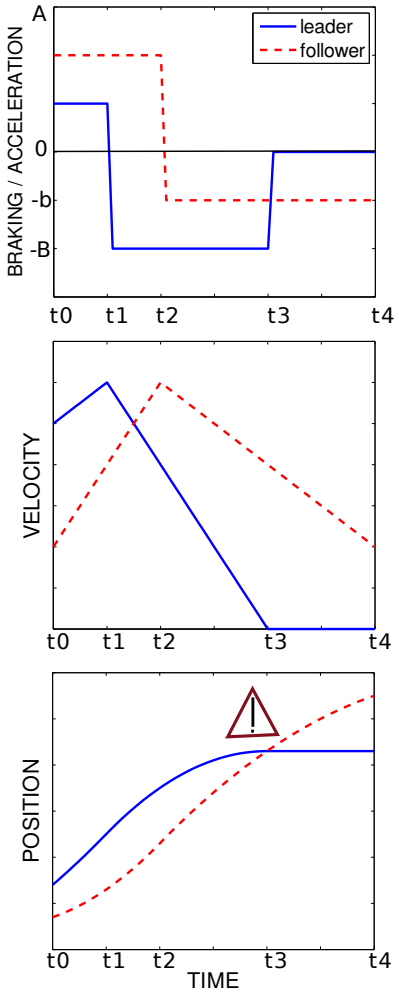

Figure 2: Local car crash

We present the entire specification of the local lane control (`llc`), consisting of the discrete control and the continuous dynamics, in Model 1. This system evolves over time, which is measured by a clock, i.e., variable t changing with slope $t' = 1$ as in (8). The differential equation system (8) formalizes the physical laws for movement, which are restricted to the evolution domain (9). Neither human drivers nor driver assistance technology are able to react immediately and each vehicle or driver will have a specific reaction time. Therefore we have a constant parameter, ε, which serves as an upper bound on the reaction time for all vehicles. We verify car control for arbitrary values of ε. Cars can react as quickly as they want, but they can take no longer than ε.

The leading car is not restricted by the car behind, so it may accelerate, coast, or brake at will. In Model 1, a_ℓ is first randomly assigned a real value, non-deterministically through (3). The model continues if a_ℓ is within the physical limits of the car's brakes and engine, i.e. between -B and A. On the other hand, f depends on the distance to ℓ and has a more restrictive set of possible moves. Car f can take some choices only if certain safety constraints about the distance and velocities are met.

Braking is allowed at all times, so a human driver may always override the automated control to brake in an emergency. In fact, braking is the only option if there is not enough distance between

Model 1 Local lane control (`llc`)

$$\texttt{llc} \equiv (ctrl; dyn)^* \tag{1}$$
$$ctrl \equiv \ell_{ctrl} \,||\, f_{ctrl}; \tag{2}$$
$$\ell_{ctrl} \equiv (a_\ell ::= *;\ ?(-B \leq a_\ell \leq A)) \tag{3}$$
$$f_{ctrl} \equiv \big(a_f ::= *;\ ?(-B \leq a_f \leq -b)\big) \tag{4}$$
$$\cup\ \big(?\mathbf{Safe}_\varepsilon;\ a_f ::= *;\ ?(-B \leq a_f \leq A)\big) \tag{5}$$
$$\cup\ \big(?(v_f = 0);\ a_f ::= 0\big) \tag{6}$$
$$\mathbf{Safe}_\varepsilon \equiv x_f + \frac{v_f^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon v_f\right) < x_\ell + \frac{v_\ell^2}{2B} \tag{7}$$
$$dyn \equiv (t := 0;\ x_f' = v_f,\ v_f' = a_f,\ x_\ell' = v_\ell,\ v_\ell' = a_\ell, t' = 1 \tag{8}$$
$$v_f \geq 0 \ \wedge\ v_\ell \geq 0 \ \wedge\ t \leq \varepsilon) \tag{9}$$

the cars for f to maintain its speed or accelerate. This is represented in (4), where there is no precondition for any force between $-B$ and $-b$.

The second possibility, (5), is that there is enough distance between the two cars for f to take any choice. This freedom is only given when (7) is satisfied. If (7) holds, then ℓ will still be safely in front of f until the controllers can react again (i.e., after they drive for up to ε time units), no matter how ℓ accelerates or brakes. This distance is greater than the minimum distance required for safety if they both brake simultaneously. The ε terms in (7) add this extra distance to account for the possibility that f accelerates for time ε even when ℓ decides to brake, which f may not notice until the next sensor update. These terms represent the distance traveled during one maximum reaction cycle of ε time units with worst-case acceleration A, including the additional distance needed to reduce the speed down to v_f again after accelerating with A for ε time units.

Now the third possibility. If f had previously chosen to brake by $a_f = -b$ then the continuous evolution dyn cannot continue with the current acceleration choices below velocity $v_f = 0$ due to constraint (9). Thus, we add the choice (6) saying that the car may always choose to stand still at its position if its velocity is 0 already.

The two cars can repeatedly choose from the range of legal accelerations. This non-deterministic repetition is represented by operator * in (1). The controllers of the two cars operate in parallel as seen in (2). Notice that the controllers are independent with respect to read and write variables (which also makes sense for implementation purposes), so in this case, parallel ($||$) is equivalent to sequential composition (;).

5.2 Verification

To verify the local lane control problem modeled in Sect. 5.1, we use a formal proof calculus for Qd$\mathcal{L}$ [Pla10]. In the local lane control problem, we want f to be safely behind ℓ at all times. To verify that a collision is not possible, we show that there is always a reasonable distance between ℓ and f; enough distance that if both cars brake instantly, the cars would not collide. We verify

this property for all times and under any condition which the system can run, so if a car can come so close to another car that even instant braking would not prevent a crash, the system is already unsafe.

For two cars f and ℓ, we have identified the following crucial relation $(f \ll \ell)$, i.e., follower f is *safely behind* leader ℓ:

$$(f \ll \ell) \equiv (x_f \leq x_\ell) \wedge (f \neq \ell) \rightarrow \left(x_f < x_\ell \wedge x_f + \frac{v_f^2}{2b} < x_\ell + \frac{v_\ell^2}{2B} \wedge v_f \geq 0 \wedge v_\ell \geq 0 \right)$$

If $(f \ll \ell)$ is satisfied, then f has a safe distance from ℓ. The formula states that, if ℓ is the leading car (i.e., $x_f \leq x_\ell$ for different cars $f \neq \ell$), then the leader must be strictly ahead of the follower, and there must be enough distance between them such that the follower can stop when the leader is braking. Also both cars must be driving forward.

The safe distance formula $(f \ll \ell)$ is the most important invariant. The system must satisfy it at all times to be verified. This is not to be confused with the definition of $\mathbf{Safe}_\varepsilon$ in the control, which must foresee the impact of control decisions for the future of ε time. For simplicity, these formulas do not allow cars to have non-zero length; however, adding the car length to x_f would eliminate this requirement.

Proposition 1 (Safety of local lane control `llc`) *If car f is safely behind car ℓ initially, then the cars will never collide while they follow the* `llc` *control model; therefore, safety of* `llc` *is expressed by the provable formula:* $(f \ll \ell) \rightarrow [\texttt{llc}](f \ll \ell)$

We proved Proposition 1 using KeYmaera, a theorem prover for hybrid systems (proof files available online [LPN11b]). A proof sketch is presented in Appendix A.1.

6 Global Lane Control

Figure 3: Lane risk

In Sect. 5 we show that a system of two cars is safe, which gives a local version of the problem to build upon. However, our goal is to prove safety for a whole highway of high-speed vehicles. The next step toward this goal is to verify safety for a single lane of n cars, where n is arbitrary and finite, and the ordering of the cars is fixed (i.e., no car can pass another). Each car follows the same control we proved safe for two cars in Sect. 5, but adding cars to the system and making it distributed has introduced new risks. It is now necessary to show, for example, if you are driving along and the car in front of you slows while the car behind simultaneously accelerates, you won't be left sandwiched between with no way to avoid a collision (as in Fig. 3).

6.1 Modeling

Because we are now looking at a lane of cars, our model will require additional features. First, we will need to represent the position, velocity, and acceleration of each car. If these variables were

Model 2 Global lane control (`glc`)

$$\texttt{glc} \equiv (ctrl^n; dyn^n)^* \tag{10}$$
$$ctrl^n \equiv \forall i : C\ (ctrl(i)) \tag{11}$$
$$ctrl(i) \equiv \big(a(i) ::= *; ?(-B \leq a(i) \leq -b)\big) \tag{12}$$
$$\cup\ \big(?\mathbf{Safe}_\varepsilon(i);\ a(i) ::= *;\ ?(-B \leq a(i) \leq A)\big) \tag{13}$$
$$\cup\ \big(?(v(i) = 0);\ a(i) ::= 0\big) \tag{14}$$
$$\mathbf{Safe}_\varepsilon(i) \equiv x(i) + \frac{v(i)^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon v(i)\right) < x(L(i)) + \frac{v(L(i))^2}{2B} \tag{15}$$
$$dyn^n \equiv (t ::= 0;\ \forall i : C\ (dyn(i)), t' = 1, t \leq \varepsilon) \tag{16}$$
$$dyn(i) \equiv x'(i) = v(i), v'(i) = a(i), v(i) \geq 0 \tag{17}$$

represented as primitives, the number of variables would be large and difficult to handle. Using only primitive variables, we cannot verify a system for any arbitrary number of cars, i.e., we could verify for, say, 5 cars, but not for any n cars. Therefore, we give each car an index, i, and use first-order variables $x(i)$, $v(i)$, and $a(i)$ to refer to the position, velocity and acceleration of car i. With these first-order variables, our verification applies to a lane of any number of cars.

Of course, the cars are all driving along the road at the same time, so we evolve the positions of the cars simultaneously along their differential equations. The acceleration, $a(i)$, of all cars is also set simultaneously in the control. We need notation for this parallel execution, so we use the universal quantifier ($\forall$) in the definition of the overall control and continuous dynamics (see (11) and (16) in Model 2). The control of all cars in the system is defined by $ctrl^n$ (11). This says that for each car i, we execute $ctrl(i)$. This control is exactly the control defined in Sect. 5 - under *any* conditions the car may brake (12); if the car is safely following its leader, it may choose any valid acceleration between $-b$ and A (13); and if the car is stopped, it may remain stopped (14). There are only two distinctions between the control introduced in `glc` and the control used in `llc` described in Sect. 5. First, we change primitive variables to first-order variables. Second, with so many cars in the system, we have to determine which car is our leader.

It is vital that every car be able to identify, through local sensors or V2V/V2I communication networks, which car is directly in front of it. It is already assumed that the sensor and communication network is guaranteed to give accurate updates to every car within time ε. We now also make the reasonable assumption that with each update, every car is able to identify which car is directly ahead of it in its lane. This may be a bit tricky if the car only has sensor readings to guide it, but this assumption is reasonable if all cars are broadcasting their positions (and which lane they occupy in the case of multiple lanes). For some car i, we call the car directly ahead of it $L(i)$, or the *leader of car* i. More formally, we assume the following properties about $L(i)$:

$$L(i) = j \equiv x(i) < x(j) \wedge \forall k : C \backslash \{i, j\}\ (x(k) < x(i) \vee x(j) < x(k))$$
$$(i \ll L(i)) \equiv \forall j : C((L(i) = j) \rightarrow (i \ll j))$$

The equation $L(i) = j$ is expanded to mean that the position of j must be ahead of the position of i, and there can be no cars between. The second formula states that for a car, i, to be safely

behind its leader, denoted $(i \ll L(i))$, we require that i should be safely behind any car which fulfills the requirements of the first equation. Each car will have at most one leader at any given time. At the end of the finite length lane, we position a stationary car. This car has no leader and therefore will never move.

The constraint $\mathbf{Safe}_\varepsilon$ from Sect. 5 has been updated to a first-order variable as well (15). It now uses $L(i)$ to identify which car is directly ahead of car i, and then determines if i is following safely enough to accelerate for ε time. This constraint is applied to all cars in the system when the individual controls set acceleration.

The continuous dynamics are the same as those described in Sect. 5, but with the added dynamics of the other cars in the system (16). Once $a(i)$ has been set for all cars by $ctrl^n$ (11), each car evolves along the dynamics of the system for no more than ε time (maximum reaction time). The position of each car evolves as the second derivative of the acceleration set by the control (17). The model requires that the cars never move backward by adding the constraint $v(i) \geq 0$. We still have a global time variable, t, that is introduced in the definition of dyn^n (16). Since $t' = 1$, all cars evolve along their respective differential equations in an absolute timeframe. Note that t is never read by the controller, thus, `glc` has no issues with local clock drift.

We model all cars in the system as repeatedly setting their accelerations as they synchronously receive sensor updates (11) and following the continuous dynamics (16). When put together and repeated non-deterministically with the * operator, these QHPs form the `glc` model (10) for global lane control. The `glc` model is easy to implement since each car relies on local information about the car directly ahead. Our online supplementary material shows a demo of an implementation of this model [LPN11b].

6.2 Verification

Now that we have a suitable model for a system of n cars in a single lane, we identify a suitable set of requirements and prove that our model never violates them. In Sect. 5, since there were only two cars on the road, it was sufficient to show that the follower car was safely behind its leader at all times. However, in this model it is not enough to only ensure safety for each car and its direct leader. We must also verify that a car is safely following all cars ahead – each car has to be safely behind its leader, and the leader of its leader, and the car in front of that car, and so on.

For example, suppose there were a long line of cars following each other very closely (they could, for instance, be in a platoon). If the first car brakes, then one-by-one the cars behind each react to the car directly in front of them and apply their brakes. In some models, it would be possible for these reaction delays to add up and eventually result in a crash [Ger97]. Our model is not prone to this fatal error, because our controllers are explicitly designed to tolerate reaction delays. Each car is able to come to a full stop no matter what the behavior of the cars in front of it (so long as all cars behave within the physical limits of their engines and brakes). To show this, we must verify that under the system controls every car is always safely behind all cars ahead until the lane ends. We do this by first defining *transitive leaders*, $L^*(i)$ as follows:

$$(i \ll L^*(i)) \equiv [k ::= i;\ (k ::= L(k))^*](i \ll k)$$

The QHP, $k ::= i;\ (k ::= L(k))^*$, continually redefines k to be the next car in the lane (until the lane ends). Because this QHP is encapsulated in $[\,]$, all states that are reachable in the program

must satisfy the formula $(i \ll k)$. In other words, starting with $(k ::= i)$, we check that i is safely behind k, or $(i \ll i)$. Next, $k ::= L(k)$, so $k ::= L(i)$, and we prove that i is safely behind k: $(i \ll L(i))$. Then we redefine k to be its leader again $(k ::= L(k))$, and we check that i is safely behind k: $(i \ll L(L(i)))$. This check is continued indefinitely: $(i \ll L(L(...\ L(i))))$. Hence the notation, $(i \ll L^*(i))$.

Proposition 2 (Safety of global lane control `glc`**)** *For every configuration of cars in which each car is safely following the car directly in front of it, all cars will remain in a safe configuration (i.e., no car will ever collide with another car) while they follow the distributed control. This is expressed by the following provable formula:*

$$\forall i : C(i \ll L(i)) \ \rightarrow\ [\texttt{glc}](\forall i : C(i \ll L^*(i)))$$

This means that as the cars move along the lane, every car in the system is safely following all of its transitive leaders.

Using Gödel's generalization rule, our proof for a lane of cars splits immediately into two branches: one which relies on the verification of the control and dynamics in the local, two car case, and one which verifies the rest of the system. These two branches are independent, and furthermore, the control and dynamics of the cars are only expanded in the verification of the local model. This is good news for two reasons. First, it keeps the resulting proof modular, which makes it possible to verify larger and more complex systems. Second, if the control or dynamics of the model are modified, only an updated verification of safety for two cars will be needed to verify the new model for the whole system. Proof details are available in Appendix A.2.

7 Local Highway Control

In Sect. 6, we verified an automated control system for an arbitrary, but constant, number of cars on a lane. Later, we will put lots of these lanes together to model highway traffic. In our full highway model, cars will be able to pass each other, change lanes, and enter or leave the highway. We first study how this full system behaves from the perspective of a single lane. When a car changes into or out of that lane, it will look like a car is appearing or disappearing in the middle of the lane: in front of and behind existing cars. Now it is crucial to show that these appearances and disappearances are safe.

If a new car cuts into the lane without leaving enough space for the car behind it, it could cause an accident. Furthermore, when two cars enter the lane simultaneously, if there are several cars between them, we must prove that there will not be a ripple effect which causes those cars between to crash (also see Fig. 1). Faithful verification must apply to all kinds of complex maneuvers and show safety for all cars in the system, not just those involved locally in one maneuver.

Our verification approach proves separate, modular properties. This allows us to compose these modular proofs and verify collision freedom for the entire system for any valid maneuver, no matter how complex, even multiple maneuvers at different places.

Model 3 Local highway control (`lhc`)

$$\texttt{lhc} \equiv (delete^*; create^*; ctrl^n; dyn^n)^* \tag{18}$$

$$create \equiv n ::= new;\ ?((F(n) \ll n) \wedge (n \ll L(n))) \tag{19}$$

$$(n ::= new) \equiv n ::= *;\ ?(\mathsf{E}(n) = 0);\ \mathsf{E}(n) ::= 1 \tag{20}$$

$$(F(n) \ll n) \equiv \forall j : C\ (L(j) = n \rightarrow (j \ll n)) \tag{21}$$

$$delete \equiv n ::= *;\ ?(\mathsf{E}(n) = 1);\ \mathsf{E}(n) ::= 0 \tag{22}$$

7.1 Modeling

We have additional challenges in modeling this new system where cars can appear and disappear dynamically. First of all, in previous sections we have used $\forall i : C$ to mean "for all cars in the system." We will now abuse this notation and take it to mean "for all cars which currently exist on this lane." (In our formal proof we use an actualist quantifier to distinguish between these situations. This technique is described in detail in another paper [Pla10].) Secondly, our model must represent what physical conditions in the lane must be met before a car may disappear or appear safely. And finally, the model must be robust enough to allow disappearances and appearances to happen throughout the evolution of the system (i.e., a car may enter or leave the lane at any time).

Recall that a car, n, has three real values: position, velocity and acceleration. Now that cars can appear and disappear, we add a fourth element: existence. The existence field is just a bit that we flip on ($\mathsf{E}(n) := 1$) when the car appears and flip off ($\mathsf{E}(n) := 0$) when the car disappears.

When we create a new car, n, we start by allowing the car to be anything. This can be written in dynamic logic as a random assignment $n ::= *$. Of course, when we look at the highway system as a whole, we won't allow cars to pop out of thin air onto the lane. This definition can be restricted to cars which already exist on an adjacent lane. However, since the choice of $*$ is non-deterministic, we are verifying our model for all possible values of n. This means that the verification required for an entire highway system will be a subset of the cases covered by this model of a single lane. Because $n ::= *$ allows n to be any car, one that exists on the lane or one that doesn't, we first must check that this "new" car isn't already on the lane. If it doesn't exist, i.e. $?(\mathsf{E}(n) = 0)$, then we can flip our existence bit to on and it will join the existing cars on this lane (20).

Now that we have defined appearance, we can define its dual: disappearance. We delete cars by choosing a car, n, non-deterministically, checking that it exists, and then flipping that bit so that it no longer exists on this lane (22). After a delete, notice that while the car ceases to exist physically on our lane, we are still able to refer to it in our model and verification as car n – a car that used to be in the lane.

A car may leave the lane at any time (assuming there is an adjacent lane which it can move into safely), but it should only be allowed to enter the lane if it is safely between the car that will be in front of it and the car that will be behind it. Because of this, when creating a car in the lane, our model will check that the car is safely between the car in front and behind. If we have a test which follows a creation of a new car, as in our definition of $create$ in (19), a new car will only appear if the test succeeds. The formula $(F(i) \ll i)$ evaluates to true if car i is safely ahead of the car

behind it. This is the dual of $(i \ll L(i))$. We define this formally in terms of $(i \ll L(i))$ as shown in (21).

The `lhc` model is identical to the `glc` model in Sect. 6, but at the beginning of each control cycle it includes zero or more car $delete$s or $create$s as shown by $delete^*$ and $create^*$ in (18). It is important to note that the verification will include interleaving and simultaneous $create$s and $delete$s since the continuous dynamics (dyn^n) are allowed to evolve for zero time and start over immediately with another $delete$ and $create$ cycle.

7.2 Verification

Now that we have a model for local highway control, we have to describe a set of requirements that we want the model to satisfy in order to ensure safety. These requirements will be identical to the requirements necessary in the global lane control. We want to show that every car is a safe distance from its transitive leaders: $\forall i : C(i \ll L^*(i))$. Because these requirements are identical to those presented in Proposition 2, the statement of Proposition 3 is identical except for the updated model.

Proposition 3 (Safety of local highway control `lhc`) *Assuming the cars start in a controllable state (i.e. each car is a safe distance from the car in front of it), the cars may move, appear, and disappear as described in the (`lhc`) model, then no cars will ever collide. This is expressed by the following provable formula:*

$$\forall i : C(i \ll L(i)) \;\rightarrow\; [\texttt{lhc}]\forall i : C(i \ll L^*(i))$$

We keep the proof of Proposition 3 entirely modular just as we did in the previous section for Proposition 2. The proof is presented in Appendix A.3.

8 Global Highway Control

So far, we have verified an automated car control system for cars driving on one lane. A highway consists of multiple lanes, and cars may change from one lane to the other. Just because a system is safe on one lane does not mean that it would operate safely on multiple lanes. When a car changes lanes, it might change from a position that used to be safe for its previous lane over to another lane where that position becomes unsafe. Lane change needs to be coordinated and not chaotic. We have to ensure that multiple local maneuvers cannot cause global inconsistencies and follow-up crashes; see Fig. 1.

8.1 Modeling

The first aspect we need to model is which lane is concerned. The quantifier $\forall i : C$, which in Sect. 7 quantified over "all cars which exist on the lane", now needs to be parametrized by the lane that it is referring to. We use the notation $\forall i : C_l$ to quantify over all cars on lane l. Likewise, instead of the existence function $\mathsf{E}(i)$, we now use $\mathsf{E}(i, l)$ to say whether car i exists on lane l. A car

could exist on some l but not on others. A car can exist on multiple lanes at once if its wheels are on different lanes (e.g., when crossing dashed lines). We use subscripted $ctrl_l^n, dyn_l^n, L_l(i), L_l^*(i)$ etc. to denote variants of $ctrl^n, dyn^n, L(i), L^*(i)$ in which all quantifiers refer to lane l. Similarly, we write $\forall l : L\ ctrl_l^n$ for the QHP running the controllers of all cars on all lanes at once.

In addition to whatever a car may do in terms of speeding up or slowing down, lane change corresponds to a sequence of changes in existence function $\mathsf{E}(i,l)$. A model for an instant switch of car i from lane l to lane l' would correspond to $\mathsf{E}(i,l) := 0; \mathsf{E}(i,l') := 1$, i.e., disappearance from l and subsequent appearance on l'. This is mostly for adjacent lanes $l' = l \pm 1$, but we allow arbitrary lanes l, l' to capture highways with complex topology. Real cars do not change lanes instantly, of course. They gradually move from one lane over to the other while (partially) occupying both lanes simultaneously for some period of time. This corresponds to the same car existing on multiple lanes for some time (studying the actual local curve dynamics is beyond the scope of this paper, but benefits from our modular hierarchical proof structure).

Gradual lane change is modeled by an appearance of i on the new lane ($\mathsf{E}(i,l') := 1$) when the lane change starts, then a period of simultaneous existence on both lanes while the car is in the process of moving over, and then, eventually, disappearance from the old lane ($\mathsf{E}(i,l) := 0$) when the lane change has been completed and the car occupies no part of the old lane anymore. Consequently, gradual lane change is over-approximated by a series of deletes from all lanes ($\forall l : L\ delete_l^*$) together with a series of appearances on all lanes ($\forall l : L\ new_l^*$). Global highway control with multiple cars moving on multiple lanes and non-deterministic gradual lane changing can be modeled by QHP:

$$\texttt{ghc} \equiv (\forall l : L\, delete_l^*;\ \forall l : L\, new_l^*;\ \forall l : L\, ctrl_l^n;\ \forall l : L\, dyn_l^n)^*$$

8.2 Verification

Global highway control `ghc` is safe, i.e., guarantees collision freedom for multi-lane car control with arbitrarily many lanes, cars, and gradual lane changing.

Theorem 1 (Safety of global highway control `ghc`) *The global highway control system (`ghc`) for multi-lane distributed car control is collision-free. This is expressed by the provable formula:*

$$\forall l : L \forall i : C_l(i \ll L_l(i)) \rightarrow \\ [(\forall l : L\, delete_l^*; \forall l : L\, new_l^*; \forall l : L\, ctrl_l^n; \forall l : L\, dyn_l^n)^*]\ \forall l : L \forall i : C_l(i \ll L_l^*(i))$$

For the proof see Appendix A.4. Note that the constraints on safe lane changing coincide with those identified in Sect. 7 for safe appearance on a lane.

9 Conclusion and Future Work

Distributed car control has been proposed repeatedly as a solution to safety and efficiency problems in ground transportation. Yet, a move to this next generation technology, however promising it may be, is only wise when its reliability has been ensured. Otherwise the cure would be worse than the

disease. Distributed car control dynamics has been out of scope for previous formal verification techniques. We have presented formal verification results guaranteeing collision freedom in a series of increasingly complex settings, culminating in a safety proof for distributed car control despite an arbitrary and evolving number of cars moving between an arbitrary number of lanes. Our research is an important basis for formally assured car control. The modular proof structure we identify in this paper generalizes to other scenarios, e.g., variations in the local car dynamics or changes in the system design. Future work includes addressing time synchronization, sensor inaccuracy, curved lanes, and asynchronous sensors.

References

[AAWB10] Matthias Althoff, Daniel Althoff, Dirk Wollherr, and Martin Buss. Safety verification of autonomous vehicles for coordinated evasive maneuvers. In *IEEE IV'10*, pages 1078 – 1083, 2010.

[BSBP03] L. Berardi, E. Santis, M. Benedetto, and G. Pola. Approximations of maximal controlled safe sets for hybrid systems. In Rolf Johansson and Anders Rantzer, editors, *Nonlinear and Hybrid Systems in Automotive Control*, pages 335–350. Springer, 2003.

[CCB+07] James Chang, Daniel Cohen, Lawrence Blincoe, Rajesh Subramanian, and Louis Lombardo. CICAS-V research on comprehensive costs of intersection crashes. Technical Report 07-0016, NHTSA, 2007.

[CT94] Wonshik Chee and Masayoshi Tomizuka. Vehicle lane change maneuver in automated highway systems. PATH Research Report UCB-ITS-PRR-94-22, UC Berkeley, 1994.

[DCH07] Thanh-Son Dao, C. M. Clark, and J. P. Huissoon. Optimized lane assignment using inter-vehicle communication. In *IEEE IV'07*, pages 1217–1222, 2007.

[DCH08] T.-S. Dao, C. M. Clark, and J. P. Huissoon. Distributed platoon assignment and lane selection for traffic flow optimization. In *IEEE IV'08*, pages 739–744, 2008.

[DHO06] Werner Damm, Hardi Hungar, and Ernst-Rüdiger Olderog. Verification of cooperating traffic agents. *International Journal of Control*, 79(5):395–421, May 2006.

[DL97] Ekaterina Dolginova and Nancy Lynch. Safety verification for automated platoon maneuvers: A case study. In Oded Maler, editor, *HART*, pages 154–170. Springer, 1997.

[Ger97] S. Germann. Modellbildung und Modellgestützte Regelung der Fahrzeuglängsdynamik. In *Fortschrittsberichte VDI, Reihe 12, Nr. 309*. VDI Verlag, 1997.

[HC02] R. Hall and C. Chin. Vehicle sorting for platoon formation: Impacts on highway entry and troughput. PATH Research Report UCB-ITS-PRR-2002-07, UC Berkeley, 2002.

[HCG03] R. Hall, C. Chin, and N. Gadgil. The automated highway system / street interface: Final report. PATH Research Report UCB-ITS-PRR-2003-06, UC Berkeley, 2003.

[HESV91] Ann Hsu, Farokh Eskafi, Sonia Sachs, and Pravin Varaiya. Design of platoon maneuver protocols for IVHS. PATH Research Report UCB-ITS-PRR-91-6, UC Berkeley, 1991.

[HTS04] Roberto Horowitz, Chin-Woo Tan, and Xiaotian Sun. An efficient lane change maneuver for platoons of vehicles in an automated highway system. PATH Research Report UCB-ITS-PRR-2004-16, UC Berkeley, 2004.

[Ioa97] P. A. Ioannou. *Automated Highway Systems*. Springer, 1997.

[JKI99] Hossein Jula, Elias B. Kosmatopoulos, and Petros A. Ioannou. Collision avoidance analysis for lane changing and merging. PATH Research Report UCB-ITS-PRR-99-13, UC Berkeley, 1999.

[JR03] Rolf Johansson and Anders Rantzer, editors. *Nonlinear and Hybrid Systems in Automotive Control*. Society of Automotive Engineers Inc., 2003.

[LL98] John Lygeros and Nancy Lynch. Strings of vehicles: Modeling safety conditions. In T.A. Henzinger and S. Sastry, editors, *HSCC*, volume 1386 of *LNCS*, pages 273–288. Springer, 1998.

[LPN11a] Sarah M. Loos, André Platzer, and Ligia Nistor. Adaptive cruise control: Hybrid, distributed, and now formally verified. In Michael Butler and Wolfram Schulte, editors, *FM*, volume 6664 of *LNCS*, pages 42–56. Springer, 2011.

[LPN11b] Sarah M. Loos, André Platzer, and Ligia Nistor. Adaptive cruise control: Hybrid, distributed, and now formally verified, 2011. Electronic proof and demo: `http://www.ls.cs.cmu.edu/dccs/`.

[LPN11c] Sarah M. Loos, André Platzer, and Ligia Nistor. Adaptive cruise control: Hybrid, distributed, and now formally verified. Technical Report CMU-CS-11-107, Carnegie Mellon University, 2011.

[Pla10] André Platzer. Quantified differential dynamic logic for distributed hybrid systems. In Anuj Dawar and Helmut Veith, editors, *CSL*, volume 6247 of *LNCS*, pages 469–483. Springer, 2010.

[SFHK04] Olaf Stursberg, Ansgar Fehnker, Zhi Han, and Bruce H. Krogh. Verification of a cruise control system using counterexample-guided search. *Control Engineering Practice*, 38:1269–1278, 2004.

[Shl04] Steven E. Shladover. Effects of traffic density on communication requirements for Cooperative Intersection Collision Avoidance Systems (CICAS). PATH Working Paper UCB-ITS-PWP-2005-1, UC Berkeley, 2004.

[Var93] P. Varaiya. Smart cars on smart roads: problems of control. *IEEE Trans. Automat. Control*, 38(2):195–207, 1993.

[WMML09] Tichakorn Wongpiromsarn, Sayan Mitra, Richard M. Murray, and Andrew G. Lamperski. Periodically controlled hybrid systems: Verifying a controller for an autonomous vehicle. In Rupak Majumdar and Paulo Tabuada, editors, *HSCC*, volume 5469 of *LNCS*, pages 396–410. Springer, 2009.

A Appendix

In this appendix we present and explain the proofs for the results presented in the main body of this paper.

A.1 Proofs for Local Lane Control

The proof of local lane control was completed in KeYmaera. To see the full proof, the file can be downloaded from `http://www.ls.cs.cmu.edu/dccs/llc.key.proof` and opened after launching KeYmaera from `http://symbolaris.com/info/KeYmaera.jnlp`. (Mathematica 7 is required, Linux is recommended.)

Safety of Local Lane Control The system in Model 1 consists of a global loop and we use $(f \ll \ell)$ as an invariant of this loop. It can be shown easily that the invariant is initially valid and implies that $(f \ll \ell)$. Proving that the invariant is preserved by the loop body $ctrl; dyn$ is the most difficult part of the proof in KeYmaera.

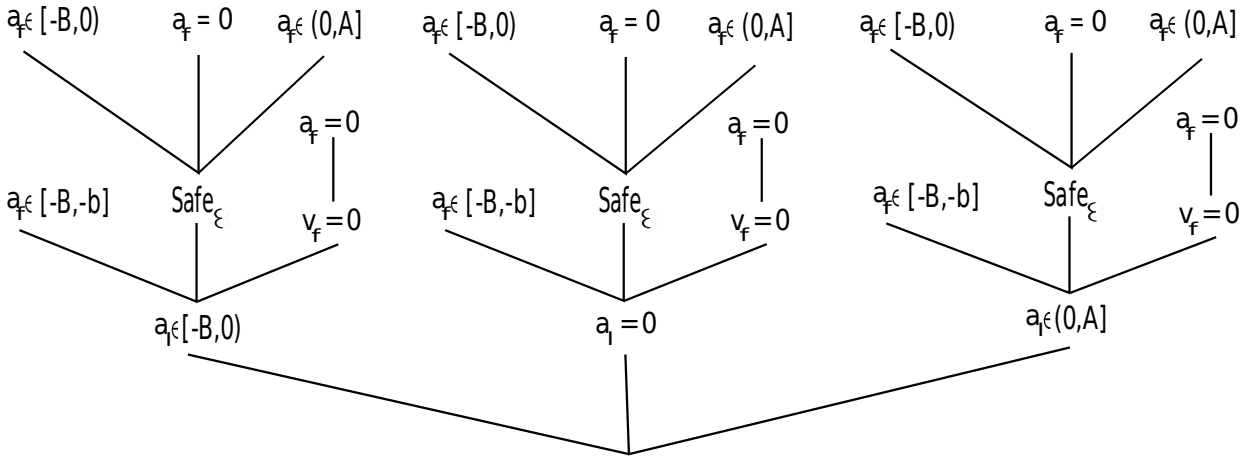

Figure 4: Proof Structure

We split the proof into multiple cases, depending on the value of a_ℓ and a_f. All cases are presented in Fig. 4. In (3) of Model 1, a_ℓ is assigned a value between $-B$ and A. In our proof, we break this assignment up into three cases: $-B \leq a_\ell < 0$, $a_\ell = 0$ and $0 < a_\ell \leq A$. For each of these three cases, there are three possibilities: it can happen that $a_f \in [-B, -b]$, that $\mathbf{Safe}_\varepsilon$ holds or that $v_f = 0$. Each possibility is represented by another subcase in the proof. If $\mathbf{Safe}_\varepsilon$ holds, the proof is further broken up into three subcases: $-B \leq a_f < 0$, $a_f = 0$ and $0 < a_f \leq A$.

There are many branches that are similar in our proof, as shown in Fig. 4. We will discuss only the left branch: when $-B \leq a_\ell < 0$, $\mathbf{Safe}_\varepsilon$ holds and $0 < a_f \leq A$. Now, the situation most susceptible to a collision is when the leader ℓ brakes with maximum braking power $-B$ and the

follower f accelerates with maximum acceleration A. We first proved that this dangerous situation is collision-free using the following insights. We identified the following useful formula that we could conclude from the assumptions in the antecedent (left of $\rightarrow$):

$$x_\ell > x_f + \frac{v_f^2}{2b} - \frac{v_\ell^2}{2B} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}t^2 + tv_f\right) \tag{23}$$

Using a lemma (which formally corresponds to a cut), we proved that this formula follows from the assumptions and then used it to prove the invariant in the remainder of the branch. The formula (23) was obtained by combining $\varepsilon \geq t$ and $\mathbf{Safe}_\varepsilon$: we applied transitivity (in the variables $\varepsilon > 0$ and $t > 0$) to the right hand side of the inequality $\mathbf{Safe}_\varepsilon$. The manual introduction of this formula was enough for KeYmaera to prove safety automatically from then on (with a small number of user interactions to simplify arithmetic reasoning and hide extra formulas). After proving that the most dangerous situation, when the leader ℓ brakes with maximum braking power $-B$ and the follower f accelerates with maximum acceleration A, is collision-free, all other situations in this subcase (left branch) can be proved collision-free. All other situations in this subcase turn out to be less dangerous, since the leader ℓ could brake with a braking power strictly bigger than $-B$, or the follower f could accelerate with an acceleration strictly smaller than A. Thus, it was possible to use a formal version of the following monotonicity argument for proving safety: if $(f \ll \ell)$ holds when the leader applies braking power $-B$, we prove that it also holds when he applies not so powerful a braking power. Similarly, if $(f \ll \ell)$ holds when the follower accelerates with acceleration A, we prove that it will hold when he applies an acceleration strictly smaller than A.

A.2 Proofs for Global Lane Control

In this section, we present a proof of Proposition 2, which was originally introduced in Sect. 6. It is restated here for convenience:

Proposition 2 (Safety of global lane control `glc`**).** *For every configuration of cars in which each car is safely following the car directly in front of it, all cars will remain in a safe configuration (i.e., no car will ever collide with another car) while they follow the distributed control, $ctrl^m$. This is expressed by the following provable formula:*

$\forall i : C(i \ll L(i)) \rightarrow [\texttt{glc}](\forall i : C(i \ll L^*(i)))$

This means that as the cars move along the lane, every car in the system is safely following all of its transitive leaders.

In proving Proposition 2, our primary objective is to keep the proof modular. In this way the control, dynamics, and verification can all be changed at the local level without affecting the global level verification. The left branch of the proof in Fig. 6 shows the early introduction of the following lemma (Lemma 1), which serves to separate the local and global proofs.

Lemma 1 (Safety of leader) *For any car, i, which is initially following the car in front of it at a safe distance, $(i \ll L(i))$, car i will remain at a safe following distance while it follows the distributed control, $ctrl^n$. That is, the following formula is provable.*

$\forall i : C(i \ll L(i)) \rightarrow [\texttt{glc}](\forall i : C(i \ll L(i)))$

In other words, as the cars move along the lane, every car will remain safely behind the car directly in front of it.

Lemma 1 follows as a corollary to Proposition 1 with two modifications: we now have an arbitrary car, i, and its respective leader, $L(i)$, instead of specific cars f and ℓ and we have control and dynamics for n cars instead of 2 cars.

In order to replace f with i and ℓ with $L(i)$ in Proposition 1, we need to guarantee that even if the leader car changes, the proof is not affected. This is very important because when we build on this to prove the safety of lane changes, the order of the cars will change frequently. By defining the lead car, $L(i)$, to be identified by a logical formula, we assure that it has all the required properties for verification, independent of a change in the cars ahead. That is to say, we don't assume that the leader is always the same car, just that it is any car which satisfies the properties of a leader. One plausible alternative would be to consider $L(i)$ to be a data field which keeps track of the leading car. However, if we were to use this approach, we would also have to go through the trouble of checking that the data field updates are always correct.

Our definitions of $ctrl^n$ and dyn^n require the control and dynamics of all cars to be executed in parallel. In our system, the control for any car, i, will only read the position and velocity fields, $x(L(i))$ and $v(L(i))$, of the car ahead and will only write its own acceleration field, $a(i)$. Because all reads and writes are disjoint, the control of one car is independent from the control of all other cars. This means that executing the car controls sequentially in any order is equivalent to executing the controls in parallel. So, without loss of generality, we may replace the universal car i in Lemma 1 with an arbitrary car, call it I (this is the logical technique of skolemization). Next we apply the hybrid control programs for all cars except I and $L(I)$. Since cars I and $L(I)$ are the only remaining cars in our formula, applying the control for the other cars has no effect and we are left with:

$(I \ll L(I)) \rightarrow [(ctrl(I); ctrl(L(I)); dyn^n)^*]((I \ll L(I)))$

Now the safety of an arbitrary car and its leader (Lemma 1) has been reduced to a form where Proposition 1 can be applied directly to prove it.

We will use Lemma 1 along with the following lemma to prove the safety of global lane distributed car control (Proposition 2).

Lemma 2 (Safety of transitive leader) *If all cars are safely following their immediate leaders, then any car, i, is also following all of its transitive leaders, $L^*(i)$:*

$\forall i : C(i \ll L(i)) \rightarrow \forall i : C(i \ll L^*(i)))$

Lemma 2 tells us that if every car is safely behind its leader, then every car is also safely behind the leader of its leader, and the car in front of that, and so on down the lane. The proof of Lemma 2 is done by induction and follows from the algebraic property that safety is transitive. A formal proof is presented in Fig. 5.

Returning to the proof of Proposition 2, we see in Fig. 6 that the property we actually need to complete the proof is

$[\texttt{glc}]\forall i : C(i \ll L(i)) \rightarrow [\texttt{glc}]\forall i : C(i \ll L^*(i)))$.

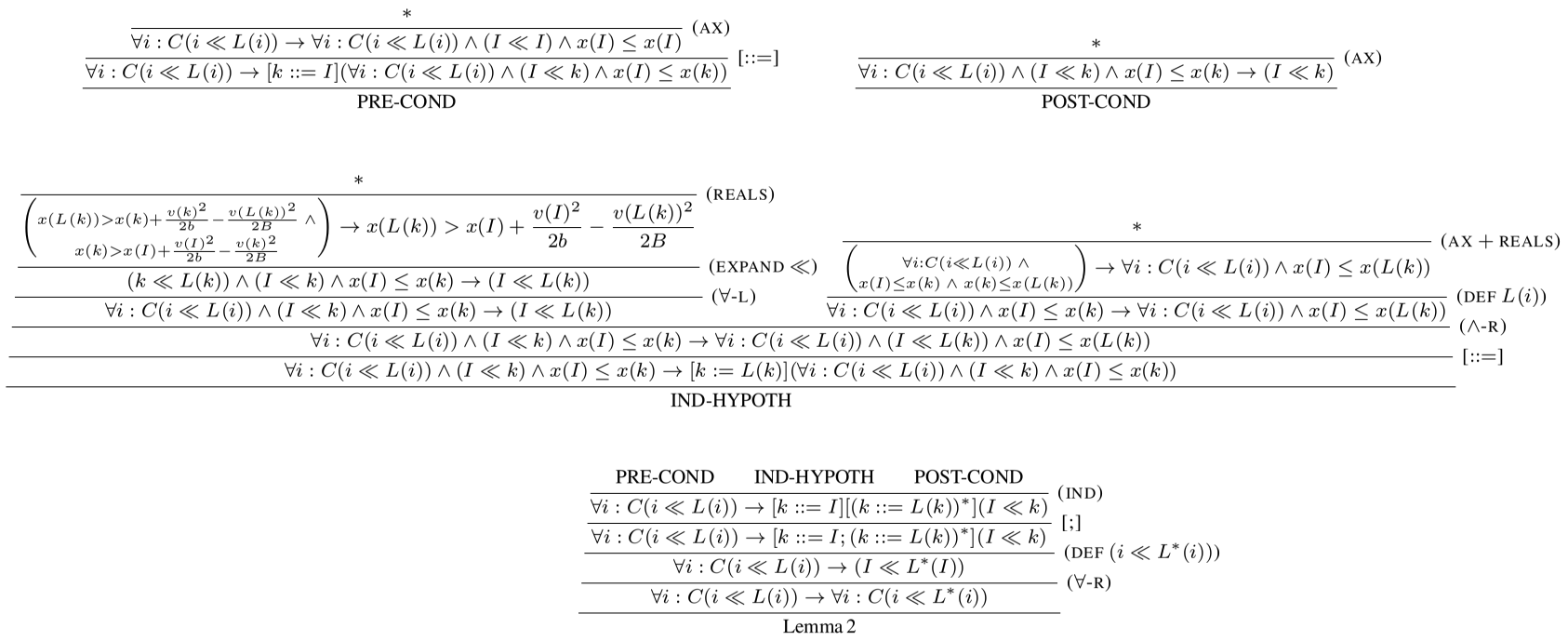

Figure 5: Proof of safety for transitive leader (Lemma 2)

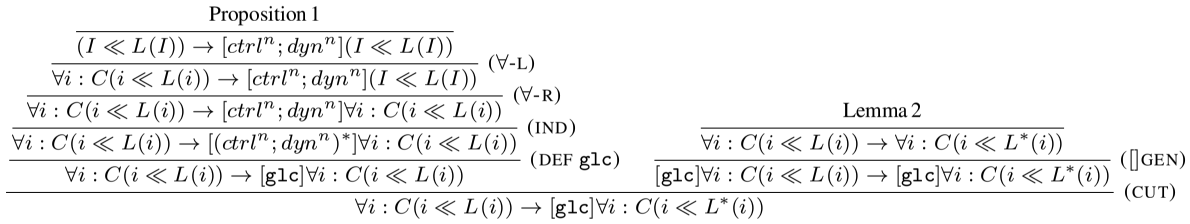

Figure 6: Proof of safety for global lane control

However, Lemma 2 is just a more general statement. If $(\phi \rightarrow \psi)$ and $[\alpha]\phi$ are valid (i.e., ϕ always holds while some QHP α is executed), then $[\alpha]\psi$ will also be valid. This is known as Gödel's generalization rule and is more formally stated as:

$$\frac{\phi \rightarrow \psi}{[\alpha]\phi \rightarrow [\alpha]\psi} \text{ ([] GEN)}$$

When looking at the complete proof structure in Fig. 6, it is important to notice that the QHP which contains the distributed control and physical dynamics of the cars is only needed in Lemma 1. Because of Gödel's generalization rule, the proof only relies on the verification of the control and dynamics in the local, two car case. It is independent of everything else. This is good news for two reasons. First, it keeps the resulting proof modular, which makes it possible to verify larger and more complex systems. Second, if the engineer who designs the system makes a change in the control or dynamics of the model later in development, under normal circumstances a new proof of safety would have to be created from scratch. However, with our modular proof structure, a new verification of safety for two cars, along with the original verification for the entire system, will be sufficient to ensure safety. The formal proof of Proposition 2 is presented in Fig. 6. (Note that we also commute $\wedge$ when we apply the ($\wedge$-R) rule.)

A.3 Proofs for Local Highway Control

To keep this proof modular, we need one crucial proof rule, ([] SPLIT):

$$\frac{\phi \rightarrow [\alpha]\phi \qquad \phi \rightarrow [\beta]\phi}{\phi \rightarrow [\alpha][\beta]\phi} \text{ ([] SPLIT)}$$

Intuitively, ([] SPLIT) makes sense as a proof rule. In the context of our distributed car control system, ϕ is the property that all cars are safely behind their transitive leaders. The QHPs, α and β, could be $delete$ and $create$ respectively. This rule says that as long as all the cars are safe before, during and after deleting some existing car, and all the cars are safe before, during and after creating a new car, then all cars are safe through the QHP which first deletes and then creates cars. Thus $[\alpha][\beta]\phi$ is valid.

More formally, ([] SPLIT) is the combination of two rules we introduced previously: CUT which was introduced in Sect. 5 and ([] GEN) introduced in Sect. 6:

$$\frac{\phi \rightarrow [\alpha]\phi \qquad \dfrac{\phi \rightarrow [\beta]\phi}{[\alpha]\phi \rightarrow [\alpha][\beta]\phi}\text{ ([] GEN)}}{\phi \rightarrow [\alpha][\beta]} \text{ (CUT)}$$

The proof of Proposition 3, presented in Fig. 7, applies ([] SPLIT) twice to split up the `lhc` model into three natural pieces: $delete^*$, new^*, and `glc`. This allows us to use the proof of Proposition 2. All that is left to prove are these two simplified statements about $delete$ and new.

$(i \ll L^*(i)) \rightarrow [delete^*](i \ll L^*(i))$
$(i \ll L^*(i)) \rightarrow [new^*](i \ll L^*(i))$

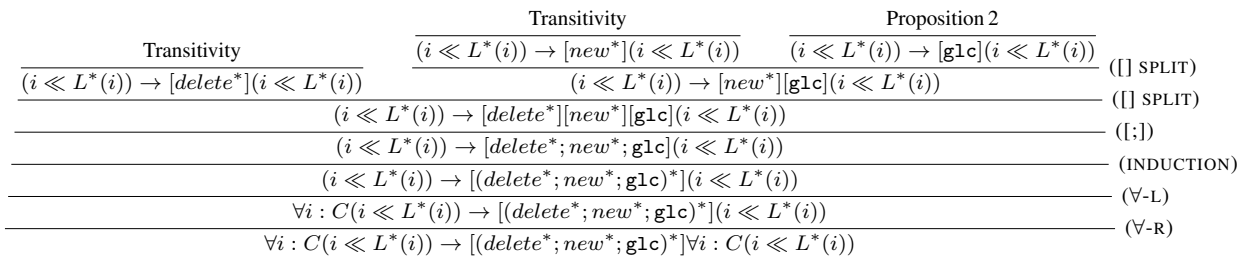

Figure 7: Proof of safety for local highway control

The first formula says that if all the cars are safely following their leaders before the $delete^*$, then all the cars will be safely following their leaders after the $delete^*$. We prove this with induction, so we must show that $(i \ll L^*(i))$ holds true after exactly one $delete$. Our definition of safety, $(i \ll j)$, is transitive. This means that when any car, n, is removed from the system, the car previously behind n (i.e., previous $F(n)$) is now safely following the car previously in front of n (i.e., previous $L(n)$).

The argument for the safety of creating a new car is equally straight forward. When a new car is allowed on the lane, it must meet certain conditions, mainly, that it is safely ahead of the car behind it and safely behind the car in front of it. Since our new car n is safely behind the car in front of it ($L(n)$) and we know that the car in front of it is safely behind all of its transitive leaders ($L^*(L(n))$), we also know that our new car is safely behind all of its own transitive leaders ($L^*(n)$). The rest of the argument follows similarly. Note that the top left branches are using the transitivity reasoning in Fig. 7. The actual proof uses lots of real arithmetic for this purpose.

A.4 Proofs for Global Highway Control

In global highway control verification, we show that the `ghc` system is collision-free. The primary extra challenge compared to the previous proofs is that we need to consider multiple lanes and prove safe switching between the lanes. What we can work with in this proof is that we have already shown in Proposition 3 that an arbitrary number of cars on one lane with arbitrarily many cars appearing and disappearing is still safe. We need to show that the cars with lane interactions work out correctly.

The proof of the global highway safety Theorem 1 is shown in Fig. 8. Theorem 1 follows from Proposition 3, which shows validity of the safety property for an arbitrary lane l. Here we make the lane l explicit in the notation of the following validity:

$$\forall i : C_l(i \ll L_l(i)) \rightarrow [(delete_l^*; new_l^*; ctrl_l^n; dyn_l^n)^*]\ \forall i : C_l(i \ll L_l^*(i))$$

This formula is an immediate corollary to Proposition 3, just by a notational change in the proof step marked by (RENAME).

In particular, the universal closure by $\forall l : L$ is still valid by $\forall$-generalization:

$$\forall l : L\big(\forall i : C_l(i \ll L_l(i)) \rightarrow [(delete_l^*; new_l^*; ctrl_l^n; dyn_l^n)^*]\ \forall i : C_l(i \ll L_l^*(i))\big)$$

This entails the formula in Theorem 1 using the fact that

$$\forall l : L\ (\phi(l) \rightarrow \psi(l))\ \text{ implies }\ (\forall l : L\ \phi(l)) \rightarrow (\forall l : L\ \psi(l))$$

by $\forall$-distribution and the fact that the formula

$$\forall l : L\ [\alpha]\phi(l)\ \text{ implies }\ [\forall l : L\ \alpha]\forall l : L\ \phi(l),$$

which we mark by (RENAME) in Fig. 8. The latter implication does not hold in general. But it does hold for the car control system, because the lane controllers satisfy the read/write independence property discussed in Sect. 6. The control of one lane is independent of the control of another lane, because we have isolated lane interaction into successive local appearance and disappearance steps. The only constraints are the appearance constraints, which are local per lane. Finally note that safety of car appearance and disappearance on the various lanes during `ghc` follows from the safety of appearance and disappearance that has been proven safe in Proposition 3.

$$\dfrac{\dfrac{\dfrac{\dfrac{Proposition\ 3}{\forall i : C_l(i \ll L_l(i)) \to [(delete_l^*; new_l^*; ctrl_l^n; dyn_l^n)^*]\ \forall i : C_l(i \ll L_l^*(i))}\ (\text{RENAME})}{\forall l : L\big(\forall i : C_l(i \ll L_l(i)) \to [(delete_l^*; new_l^*; ctrl_l^n; dyn_l^n)^*]\ \forall i : C_l(i \ll L_l^*(i))\big)}\ (\forall\text{GEN})}{\forall l : L \forall i : C_l(i \ll L_l(i)) \to \forall l : L[(delete_l^*; new_l^*; ctrl_l^n; dyn_l^n)^*]\ \forall i : C_l(i \ll L_l^*(i))}\ (\forall\text{DIST})}{\forall l : L \forall i : C_l(i \ll L_l(i)) \to [(\forall l : L\ delete_l^*; \forall l : L\ new_l^*; \forall l : L\ ctrl_l^n; \forall l : L\ dyn_l^n)^*]\ \forall l : L \forall i : C_l(i \ll L_l^*(i))}\ (\text{INDEP})$$

Figure 8: Proof of safety for global highway control